

# BEST AVAILABLE COPY

Application No.: 09/816,752

Docket No.: SCEI 3.0-058

## REMARKS

This reply is in response to the Official Action mailed December 14, 2004 rejecting all of the pending claims, namely, claims 1-38. A petition for a one-month extension of time is enclosed herewith. Claims 1-38 are again presented for the Examiner's consideration. Claims 1 and 20 are independent.

The undersigned appreciates the acknowledgement of the references cited in the Information Disclosure Statements. In the Information Disclosure Statement dated June 1, 2004, reference CA to Suzuki was not initialed because no translation was provided. Applicants submit herewith a translation of this Suzuki reference for consideration.

Claims 1-5, 15, 20-22, and 35-38 were rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,581,777 ("Kim"). Applicants respectfully traverse the rejection.

Kim is directed to transferring data within a parallel processor system. The processor system includes processor clusters 40 having multiple "substantially identical processor elements 50.00-50.15, which include respective processors 49.00-49.15 connected to respective stage registers 52.00 through 52.15...Stage registers 52.00-2.15 are connected to cluster memory 54 by common data bus 53." (Fig. 2; Col. 5, lns. 33-39.) An array control unit (ACU) 20 includes a transfer controller 62 and a processor controller 56 connected to the stage registers 52 and the cluster memory 54 for each processor cluster 40. (See FIG. 2.) A memory output (read) operation is explained in Kim as follows:

Processor controller 56 instructs the transfer controller 62 over line [68] to initiate the memory read.... [T]he transfer controller 62 signals "busy" on line 69, instructs the memory 54 over line 66 to

output data, and instructs stage registers 52.00-52.15 over line 64 to receive data, one stage register at a time. Processor controller 56 instructs processors 49.00-49.15 to continue parallel processing operations ... provided they are not incompatible with the "busy"0 signal on line 69. When data from memory 54 is loaded into participating ones of the stage registers 52.00-52.15, the transfer controller 62 interrupts the processor controller 56 on line 69, transfers data from the stage registers 52.00-52.15 to the processors 49.00-49.15 ... and releases the interrupt of processor controller 56.

(Col. 6, lns. 32-50; emphasis added.)

However, while *Kim* discloses transferring data from the memory 54 to the stage memories 52, *Kim* does not teach or suggest, for instance, "a main memory for storing programs and data associated with said programs," "a plurality of first processing units for processing said programs and said associated data," or "a second processing unit for controlling said processing of said programs and said associated data by said first processing units, said second processing unit being operable to direct any one of said first processing units to process one of said programs by directing the transfer of said one program and data associated with said one program from said main memory to the local memory exclusively associated with said one first processing unit and instructing said one first processing unit to initiate processing of said one program, said one first processing unit thereafter processing said one program and said data associated with said one program from said local memory" as required by independent claims 1 and 20 (emphasis added).

Thus, for at least these reasons, Kim does not teach or suggest all of the limitations of the independent claims. Claims 2-5, 15, 21-22 and 35-38 depend from independent claims 1 and 20, respectively, and contain all the limitations thereof as well as other limitations that are neither disclosed nor suggested by these references. Applicants respectfully request, therefore, that the rejection based on Kim be withdrawn.

Claims 1-3 and 16-20 were rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Publication No. 2002/0078285 ("*Hofstee*"). Applicants respectfully traverse the rejection.

*Hofstee* discloses a system and method for executing remote procedure calls in a multiprocessor system. Multiple processing elements 20A-20D are connected to a shared memory 10. (See FIG. 1.) The processing elements include attached processing units (APUs) 230. (See FIG. 2.) The Examiner asserts that each APU inherently includes a local memory. In particular, the Examiner states that "each said first processing unit includ[es] a local memory exclusively associated with said first processing unit (local memory is inherent for copying data to/from the APU as described in paragraph 0028)." (Official Action at 5.) *Hofstee*, however, provides no disclosure of a local memory being associated with an APU. In particular, *Hofstee* provides no teaching or suggestion of a set of local memories wherein "each said first processing unit include[s] a local memory exclusively associated with said first processing unit," as required by independent claim 1, or wherein "each of said local memories being exclusively associated with a different one of said first processing units," as required by claim 20 (emphasis added).

Because *Hofstee* does not teach or suggest exclusively associating a local memory with one first processing unit, the reference cannot teach or suggest "directing the transfer of

said one program and data associated with said one program from said main memory to the local memory exclusively associated with said one first processing unit and instructing said one first processing unit to initiate processing of said one program, said one first processing unit thereafter processing said one program and said data associated with said one program from said local memory" as required by independent claims 1 and 20.

Thus, for at least these reasons, *Hofstee* does not teach or suggest all of the limitations of the independent claims. Claims 2-3 and 16-19 depend from independent claim 1 and contain all the limitations thereof as well as other limitations that are neither disclosed nor suggested by *Hofstee*. Applicants respectfully request, therefore, that the rejection based on *Hofstee* be withdrawn. In addition, *Hofstee* cannot be applied against Applicants' claims under 35 U.S.C. § 103(a) in view of 35 U.S.C. § 103(c)(2), which states:

(2) For purposes of this subsection, subject matter developed by another person and a claimed invention shall be deemed to have been owned by the same person or subject to an obligation of assignment to the same person if -

(A) the claimed invention was made by or on behalf of parties to a joint research agreement that was in effect on or before the date the claimed invention was made;

(B) the claimed invention was made as a result of activities undertaken within the scope of the joint research agreement; and

(C) the application for patent for the claimed invention discloses or is amended to disclose the names of the parties to the joint research agreement.

*Hofstee* is owned by International Business Machines Corporation. As indicated by the amendment to the

specification, the inventions disclosed and claimed in this application were made as a result of activities undertaken within the scope of a joint research agreement among Sony Computer Entertainment Inc., Toshiba Corporation and International Business Machines Corporation. The joint research agreement was executed March 9, 2001, which was before the claimed invention was made. This information is provided in the attached Statement Pursuant to 35 U.S.C. § 103(c)(2), which has been signed on behalf of all of the assignees of the instant application.

Claims 1-3, 6-8, 10-13, 16-20, 23-28 and 30 were rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 5,487,146 ("Guttag"). Applicants respectfully traverse the rejection.

Guttag is generally directed to controlling the addresses of memory access, for example in a multiprocessor system. The system includes multiple digital image/graphics processors 71-74, which are controlled by a master processor 60, as well as a transfer controller 80, for mediating access to system memory. (See FIG. 2; see also col. 10, lns. 6-20.) The system also includes an additional memory 9, as well as random access memories 10 and 20.

The Examiner asserts that the memory 9 is "a main memory for storing programs and data associated with said programs," and that the memory 20 is "exclusively associated with said first processing unit." (Official Action, pg. 7, ¶ 23.) Applicants respectfully disagree.

According to Guttag, "[m]ultiprocessor integrated circuit 100 may be controlled either in wholly or partially by a program stored in the memory 9. This memory 9 may also store various types of graphic image data." (Col. 9, lns. 18-21.) However, independent claims 1 and 20 each require "a main memory for storing programs and data associated with said programs"

(emphasis added). There is no such teaching or suggestion in *Guttag*. There is simply no indication whatsoever that the graphic image data in *Guttag* is associated with the program for controlling multiprocessor integrated circuit 100.

With regard to the memory 20, *Guttag* states that "[r]andom access memory 20 may be accessed by master processor 60 and each of the digital image/graphics processors 71, 72, 73 and 74." (Col. 10, lns. 27-29.) As seen in FIG. 2, crossbar 50 is a necessary part of the system 100 to enable data transfers and data access among the different processors. (See col. 11, line 62 to col. 12, line 31.) Thus, the memory 20 cannot be "a local memory exclusively associated with said first processing unit" as recited in claim 1, or meet the limitation of "each of said local memories being exclusively associated with a different one of said first processing units," as required by claim 20.

Thus, for at least these reasons, *Guttag* does not teach or suggest all of the limitations of the independent claims. Claims 2-3, 6-8, 10-13, 16-19, 23-28 and 30 depend from independent claims 1 and 20, respectively, and contain all the limitations thereof as well as other limitations that are neither disclosed nor suggested by *Guttag*. Applicants respectfully request, therefore, that the rejection based on *Guttag* also be withdrawn.

Claims 14 and 31-34 were rejected under 35 U.S.C. § 103(a) as being obvious over *Guttag* in view of U.S. Patent No. 6,467,012 ("Alvarez"). Claim 15 was rejected under 35 U.S.C. § 103(a) as being obvious over *Guttag* in view of U.S. Patent No. 5,940,870 ("Chi"). Claims 9 and 29 were rejected under 35 U.S.C. § 103(a) as being obvious over *Guttag* in view of U.S. Patent No. 5,497,465 ("Chin"). Claims 4-5, 21-22, and 35-38 were rejected under 35 U.S.C. § 103(a) as being obvious over

Guttag in view of Kim. Applicants respectfully traverse each of these rejections.

Claims 4-5, 9, 14-15, 21-22, 29 and 31-38 depend from independent claims 1 and 20, respectively, and contain all the limitations thereof as well as other limitations. As stated above, Guttag does not teach or suggest the subject matter of independent claims 1 and 20. The other references identified above do not overcome the deficiencies of Guttag. Applicants respectfully request, therefore, that these rejections be withdrawn.

As it is believed that all of the rejections set forth in the Official Action have been fully met, favorable reconsideration and allowance are earnestly solicited. If, however, for any reason the Examiner does not believe that such action can be taken at this time, it is respectfully requested that he telephone Applicants' attorney at (908) 654-5000 in order to overcome any additional objections which he might have.

If there are any additional charges in connection with this requested amendment, the Examiner is authorized to charge Deposit Account No. 12-1095 therefor.

Dated: March 23, 2005

Respectfully submitted,

By Andrew T. Zidel  
Andrew T. Zidel  
Registration No.: 45,256  
LERNER, DAVID, LITTENBERG,  
KRMHOLZ & MENTLIK, LLP  
600 South Avenue West  
Westfield, New Jersey 07090  
(908) 654-5000  
Attorney for Applicant

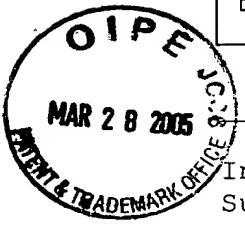
LD-4471  
547333\_1

I hereby certify that this correspondence is being deposited with the U.S. Postal Service with sufficient postage as First Class Mail, in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date shown below.

Dated: March 23, 2005 Signature: Andrew T. Zidel  
(Andrew T. Zidel)

Docket No.: SCEI 3.0-058  
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

  
In re Patent Application of:  
Suzuoki et al.

Application No.: 09/816,752

: Group Art Unit: 2154

Filed: March 22, 2001

: Examiner: A.C. Perez  
Daple

For: PROCESSING MODULES FOR COMPUTER  
ARCHITECTURE FOR BROADBAND  
NETWORKS

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**STATEMENT PURSUANT TO 35 U.S.C. § 103(c) (2)**

Dear Sir:

We hereby declare that:

1. Sony Computer Entertainment Inc. ("SCEI"), International Business Machines Corp. ("IBM"), and Toshiba Corporation ("Toshiba") are the assignees of the entire interest of the claimed invention of the above-identified pending U.S. Patent Application No. 09/816,752 ("the '752 application") filed on March 22, 2001 and which is entitled PROCESSING MODULES FOR COMPUTER ARCHITECTURE FOR BROADBAND NETWORKS.

2. The claimed invention was made by or on behalf of SCEI, IBM and Toshiba based upon a joint research agreement that was in effect on or before the date the claimed invention was made.

3. The claimed invention was made as a result of activities undertaken within the scope of the joint research agreement.

4. U.S. Patent Pub. No. 2002/0078285 ("Hofstee"), filed December 14, 2000 and published June 20, 2002, is owned by IBM.



Application No.: 09/816,752

Docket No.: SCEI 3.0-058

5. The application for patent for the claimed invention has been amended to disclose the names of the parties to the joint research agreement.

Sony Computer Entertainment, Inc.

By: \_\_\_\_\_

Name: \_\_\_\_\_

Title: \_\_\_\_\_

Date: \_\_\_\_\_

International Business Machines Corporation

By: Diana Roberts Gerhardt

Name: Diana Roberts Gerhardt

Title: Senior Counsel, IP LAW

Date: March 16, 2005

Toshiba Corporation

By: \_\_\_\_\_

Name: \_\_\_\_\_

Title: \_\_\_\_\_

Date: \_\_\_\_\_



Application No.: 09/816,752

Docket No.: SCEI 3.0-058

MAR 28 2005 995. The application for patent for the claimed invention has been amended to disclose the names of the parties to the joint research agreement.

Sony Computer Entertainment, Inc.

By: Tetsuya Kobayashi

Name: Tetsuya Kobayashi

Title: Vice President

Date: March 15, 2005

International Business Machines Corporation

By: \_\_\_\_\_

Name: \_\_\_\_\_

Title: \_\_\_\_\_

Date: \_\_\_\_\_

Toshiba Corporation

By: \_\_\_\_\_

Name: \_\_\_\_\_

Title: \_\_\_\_\_

Date: \_\_\_\_\_

5. The application for patent for the claimed invention has been amended to disclose the names of the parties to the joint research agreement.

Sony Computer Entertainment, Inc.

By: \_\_\_\_\_

Name: \_\_\_\_\_

Title: \_\_\_\_\_

Date: \_\_\_\_\_

International Business Machines Corporation

By: \_\_\_\_\_

Name: \_\_\_\_\_

Title: \_\_\_\_\_

Date: \_\_\_\_\_

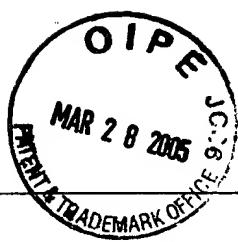
Toshiba Corporation

By: Takashi Abe

Name: Takashi Abe

Title: Patent Specialist

Date: 05/3/14



Computer Science Magazine  
[Publication information]

**January  
2000**  
Vol. 32,  
No. 1

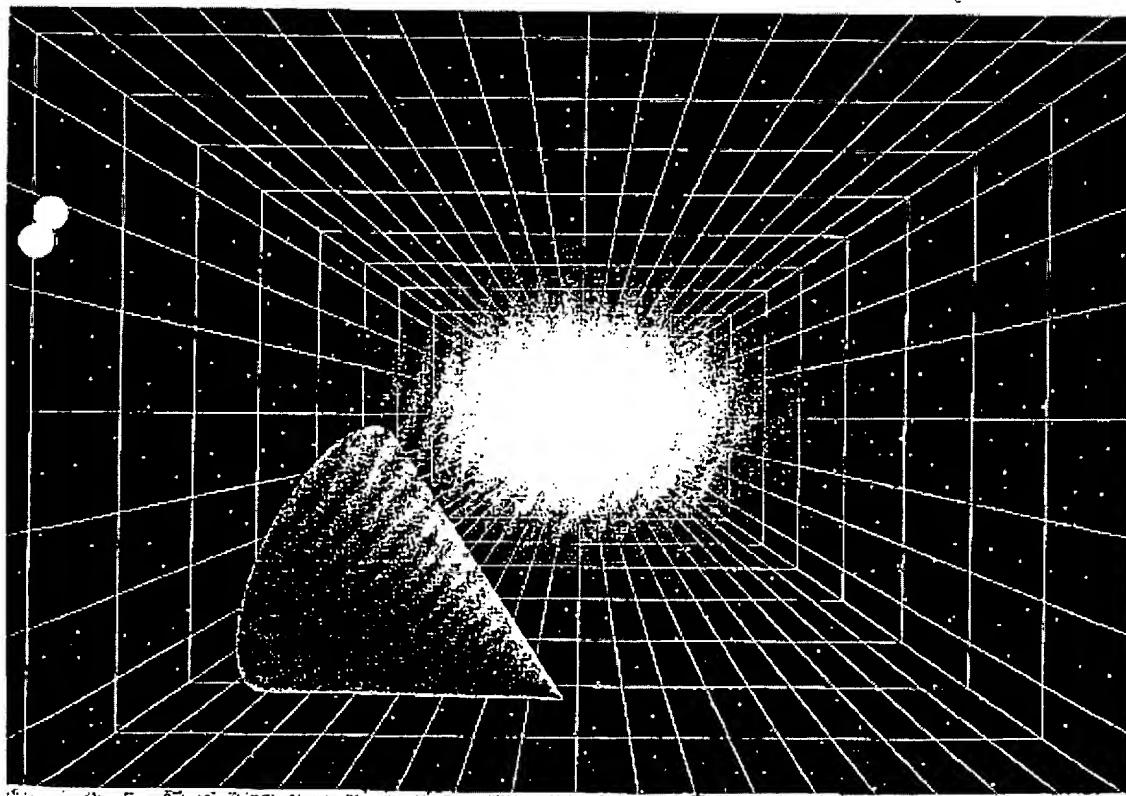
# bit

## Special Edition

### CPU Trends

[list of articles not related to attached]

[Received stamp dated  
12/27/99]



# The “Emotion Engine”

## Microprocessor for the PlayStation 2

Masakazu Suzuoki

### (1) Introduction

It is being said that finally the day is approaching when the computer will be part of the home in a big way. When seen for home use, the cost-performance ratio of computers has approached the level where computers will be accepted by the general consumer. From this idea, chips built around a microprocessor that have built-in peripheral devices (system LSI chips) are being developed and announced one after another. Meanwhile, general-purpose microprocessors are also expanding their instructions for video and audio processing, focused on home applications. For both, the goal is next-generation home information appliances that combine audio-visual and computer functions, and taking its position as one of these is a main processor known as the Emotion Engine (EE), which was developed for the PlayStation 2 next-generation video game machine by Sony Computer Entertainment.

This processor specializes in real-time information processing, mainly in video games, so it has a unique architecture that is distinguished from that of general microprocessors. Here

we outline its architecture and take an overview of the video game applications that are demanded of it.

### (2) What is needed?

First, what is required of the nature of a home computer (console) represented by a video game machine? Because a home computer is for those who do not have a deep knowledge of computers and is intended for entertainment, what is required of it is different from what is needed for an ordinary computer.

#### 2.1 The latest process

First, the functions and performance of a console must not change during the lifetime of the product. Changes in the specifications for a console, even if to improve its performance, are good for neither developers nor users. In the world of consumers, all hardware and software (titles) must be perfectly one-to-one. This is like insisting that all CDs must be playable on all CD players. Guaranteeing this is even an obligation of the platform folder. If this effort is neglected, success will not be achieved in the consumer market. Since no changes can be made midway, the most advanced

technology possible must be adopted from the first time a product goes on the market.

#### 2.2 High calculation capacity

Next, the user does not operate a console for work or as a duty. To attract the user, it is important to have something to “grab” him. The graphics must strike the eye and capture the heart. But impressive high-quality computer graphics takes lots of computation.

In addition, thinking and reasoning and the simulation of physical phenomena are essential for true quality entertainment. What gives depth to games and applications are the actions of characters that change adaptively with the situation, supported by computation rather than referring to previously prepared tables. If all the computation devices are used up in calculating coordinates for graphics, the extra computation resources for calculating behavior and action will be depleted, and the application will become simply something that is interesting to look at. Games that start off splendidly but have no content are the most despised. To avoid this too, it is necessary to have powerful, more-than-enough computing power (especially the capacity for floating-point computation).

### (3) What is lacking?

Thus a console must exploit to the limit the latest semiconductor manufacturing technology (processing technology), have ample computing power, and maintain a long useful lifetime. But progress in processing technology is not universally effective for all the factors of a system. The bottleneck factors change with the evolution of processing; for example, as the frequency within a chip improves, memory access problems might be created. Conversely, discerning where the bottlenecks are reveals where resources must be concentrated.

#### 3.1 Capacity for computing coordinates

Looking into the matter, progress in processing technology turns out to be a matter of reducing the size of (miniaturizing) the logic circuits (gates) that make up a large-scale-integrated circuit. Miniaturization leads to improved operating frequency and more gates. In particular, the number of gates is proportional to the square of the process rules, so the degree of improvement is dramatic. Increasing the total number of gates

makes it simple to align many circuits of the same kind on a chip. In graphics processing, parallelization goes along well with pixel processing, in which simple processing is repeated many times. In addition, if memory is put onto the left-over part (in a mixed DRAM), then the bandwidth to the frame buffer can be expanded. Because of this, in modern processing it is relatively easy to improve picture-drawing performance (the speed at which pixels are written to). Then the corresponding geometry processing (coordinate transformation processing) becomes relatively inadequate. In other words, even if it becomes possible to draw many polygons to the frame buffer, this time it is no longer possible to compute all their vertices in time.

#### 3.2 Memory access

The memory system is more critical than the geometry bottlenecks.

First, the capacity of the high-speed memory is no longer adequate. Raising the operating frequency will not increase the memory capacity, so memory becomes more valuable to the extent that the amount of computation increases in proportion to the frequency. Thus once the macro pipeline between processors becomes congested, the intermediate buffer memory will immediately become full.

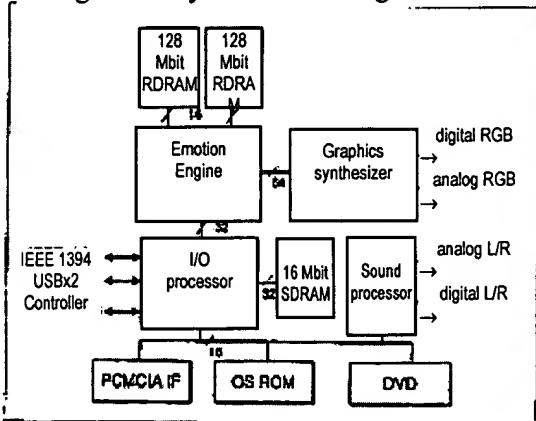
When this happens, from the standpoint of data flow, the upstream processor must wait until a buffer becomes empty, with the result that there is no increase in performance.

Moreover, the random access speed will not be sufficient. Even if the processing is made finer, the physical performance of a DRAM cell will not improve much. Even given that a protocol is introduced in the memory bus, in the end, copying from a memory cell to a sense amplifier constitutes a bottleneck, so random access that straddles columns will be no faster. The penalty for a page break (the overhead cycles when straddling columns or banks) seen in terms of cycles beggars the imagination, and already high-speed processors cannot make direct contact with the external main memory.

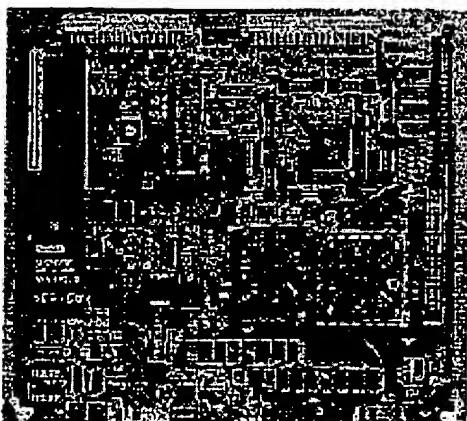
#### 3.3 Context switching

The pipelines inside processors that operate at a high frequency are unavoidably complicated. This makes the internal state (context) of the processor complicated as well, and the cost of context switching when switching from one process (thread) to another can no longer be ignored. In a multi-thread real-time system, interrupts cannot be avoided, but

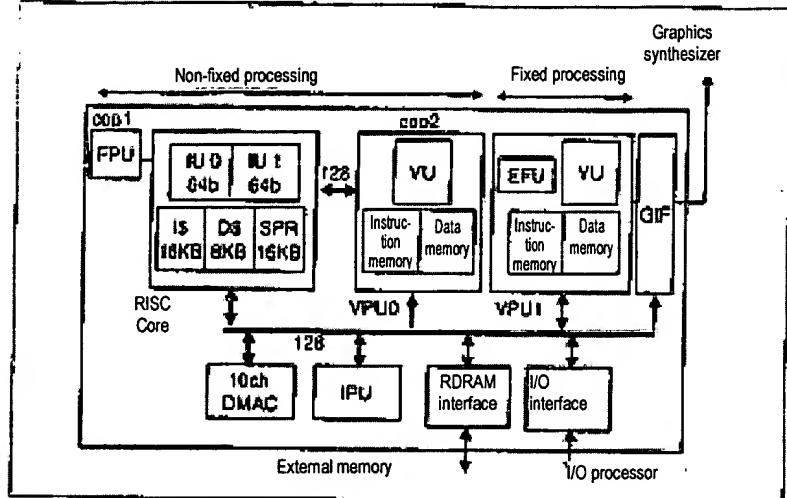
<Figure 1> System block diagram



<Photo 1> Circuit board



<Figure 2> Block diagram of the Emotion Engine (EE)



context switching (cutting off a pipeline) caused by an interrupt is completely unpredictable from the interrupted side, which creates properties that are far worse than with a cutoff due to branching. In addition, context switching disturbs the coherence (consistency) of the cache memory and lowers the cache's hit rate. Thus for a modern microprocessing unit (MPU), context switching is completely undesirable, and it is advisable to keep context switching few and far between.

#### (4) Architecture

Thought of in this way, one realizes that as necessary requirements for a next-generation console, it is important to have

<Table 1> Main specifications

Core	MIPS III 2-issue [sic: 2-way? ] superscalar 128b multi-media instruction expansion
Instruction cache	16 KB 2-way
Data cache	8 KB 2-way
Scratchpad memory	
Data bus	128b (@ 150 MHz)
FPU	1 FMAC + 1 FDIV
VPU	5-way 64b VLIW
VPU0	4 FMAC + 1 FDIV
VPU1	4 FMAC + 1 FDIV
EFU	1 FMAC + 1 FDIV
IPU	MPEG2 decoder accelerator
DMAC	10 channels
System clock	300 MHz

powerful computation capacity, be attentive to the memory system, and keep context switching to a minimum. The EE [Emotion Engine] was designed with these points in mind. A block diagram of the PlayStation 2 is shown in Figure 1, and Photo 1 shows its circuit board. The EE is a control processor positioned in the center of the system. A block diagram of the EE is shown in Figure 2.

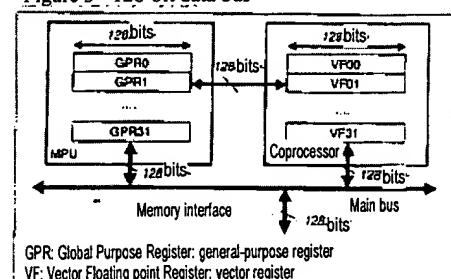
Integrated into the LSI chip with a 128-bit MPU core are two vector units (Vector Processing Units: VPU0, VPU1) and an image processing engine (Image Processing Unit: IPU), and the units are connected with a 10-channel DMA controller (DMAC). These are controlled by a core processor (MPU) that is compliant with MIPS-III (including some MIPS-IV instructions). A direct RAM bus

(RDRAM) is adopted as the external memory. Each unit independently has a floating-point multiply-adder circuit (FMAC) and floating-point divider (FDIV), each VPU has four FMACs and one FDIV, and the MPU has one FMAC and one FDIV as floating-point coprocessors (FPU). Furthermore, VPU1 also has one FMAC and one FDIV as an elementary function unit (EFU). Thus a total of ten FMACs and four FDIVs are integrated into the system, and they can share the main memory via the DMAC and can operate independently or in parallel. The basic specifications of the system are listed in Table 1.

#### 4.1 MPU

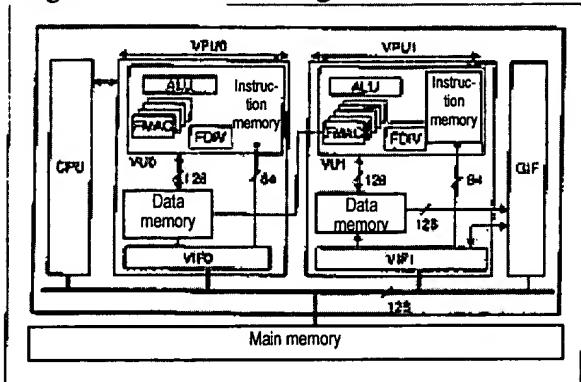
The MPU is a 2-way superscalar MIPS-III-based microprocessor, and its data bus has an all-128-bit width. The 32 general purpose registers (GPR: Global Purpose Register) all have a width of 128 bits, and they are connected via the 128-bit coprocessor bus to the coprocessor (VPU0), which also has a 128-bit width (Figure 3). The 128-bit registers are partitioned into 8x16, 16x8, and 32x4 fields, and instructions (multi-media instructions) have been added to perform the same processing on each field. The MPU has a 16-KB instruction cache and an 8-KB data cache, and these can be accessed in a single cycle. In addition, it has a 16-KB scratchpad RAM (SPR) as a high-speed local memory.

<Figure 3> 128-bit data bus



GPR: Global Purpose Register; general-purpose register  
VF: Vector Floating point Register; vector register

<Figure 4> VPU block diagram



#### 4.2 VPU

Each VPU has four FMACs and an FDIV and is controlled by a microprogram. A block diagram of a VPU is shown in Figure 4, and its basic performance is given in Table 2.

The program takes the form of 64-bit long word instructions (LIW), and each instruction consists of single instruction multiple data (SIMD). LIW is a function in which multiple predetermined instructions can be executed in one cycle, and a VPU can simultaneously execute two 32-bit instructions. Also, SIMD is a composition that can perform the same operation on multiple items of data simultaneously, and with a VPU it is possible with a single instruction to operate four FMACs simultaneously and perform four-dimensional vector calculations in a single cycle (throughput) (Figure 5).

#### 4.3 IPU

Two-dimensional images such as moving pictures and texture images are compressed and expanded by the IPU. The expansion method is by MPEG2 and its subsets. As post-processing, vector quantization (VQ) can be done on data that has been decoded. The IPU does not just expand

texture data but also has the function of MPEG2 decoding. In decoding, the main memory is recruited for use as a working memory. The data flow in decoding is pictured in Figure 6.

the DMAC. Control of the DMAC is done only by the MPU, so in the end the MPU is informed of all the main bus transactions. Although read-writes between local memories corresponding to each unit are done individually, there are no transactions that appear on the main bus that the MPU does not know about. Because of this, control of multiple programmable units is made easy.

A unit's local memory is logically dual-port, and can be accessed even during a DMA transfer to the local memory. This allows transfers and calculations to be executed simultaneously. For example, the MPU has a local buffer called a scratchpad RAM (SPR), but, as shown in Figure 8, by making the SPR a double buffer, transfer processing between the memory and SPR and MPU calculation processing can be done in parallel.

But the data to be sent to a unit's local memory is not always located in continuous addresses. In graphics applications especially, often there are many data structures that have a list structure. Every time the nodes of a list are traversed, an interrupt is made to the MPU and the DMAC transfer address is rewritten, so the result is

<Table 2> VPU specifications

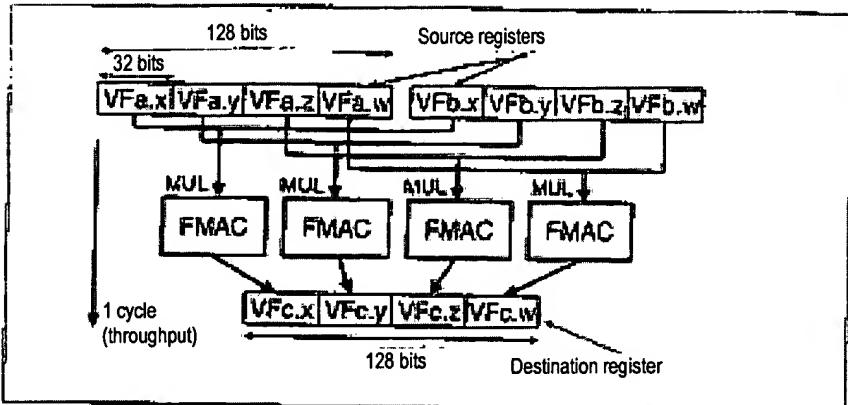
	VPU0	VPU1
Vector registers	128 bits x 32	
Integer registers	16 bits x 16	
FMAC	single-precision floating-point x 4	
Multiply add	1 cycle	
Minimum, maximum	1 cycle	
FDIV	single-precision floating-point x 1	
Division	7 cycles	
Square root	7 cycles	
Reciprocal square root	13 cycles	
Instruction bus	64 bits	
Coprocessor bus	128 bits	
Instruction memory	4 KB	16 KB
Data memory	4 KB	16 KB

#### (5) Data flow

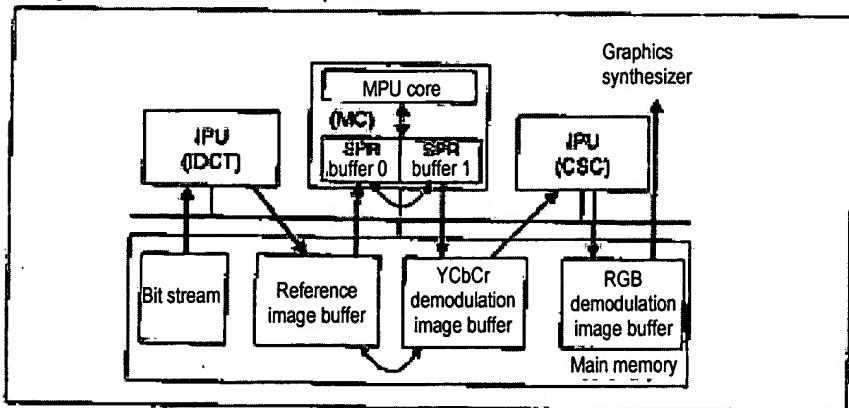
Data is passed between the various units via the main main [sic; main memory?]. But when each unit issues a read-write access directly to the main memory, what often happens from the standpoint of the main memory is a random transaction, which it deals with poorly. Thus each unit has its own local high-speed working memory, where temporary collection for detailed access takes place. As a result, most access to the memory system is bussed access done in batch units, and an improvement in throughput can be expected (Figure 7).

Actually, except for the MPU, the units act as bus masters and have no authority to access the memory directly on their own. To copy data to a local memory, from the main memory to a unit, is the role of the DMA controller (DMAC). Similarly, synchronous control between units is done all by

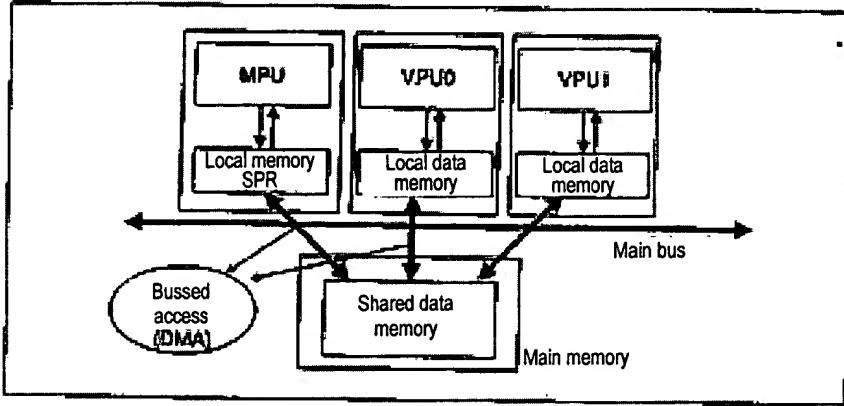
<Figure 5> 128-bit parallel processing



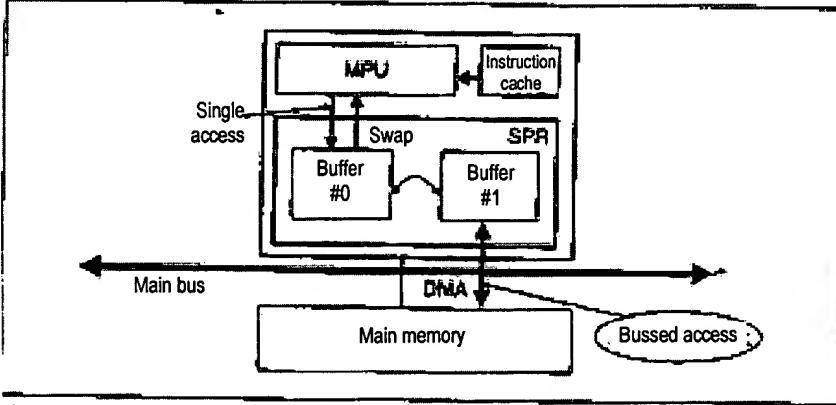
<Figure 6> MPEG2 video decoding



<Figure 7> Composite shared memory model



<Figure 8> Scratchpad memory



an enormous number of interrupts and the occurrence of context switching. Thus if it is a simple, single-direction list, the DMAC has a way to traverse it on its own. An example of this data structure is shown in Figure 9. In this example, matrix and vertex data that has been placed in discrete locations of the memory are linked by a list, and the DMAC, while traversing the list, transfers its content to the VPU. DMAC transfer is cut up into 8-qword (128-byte) units called slices, and upon request, the bus authorization is returned to the MPU. Because of this, the MPU never is made to wait more than necessary until the bus becomes empty, even during a DMA transfer.

One advantage of this configuration is that no unit needs to know about the address space in the main memory. Information about the main memory address of the data that is sent is not included in the VPU program. The VPU microprogram code can order things around in the same way as with ordinary data, and does not need to relink to fit changes in the main program.

(6) Solved like this

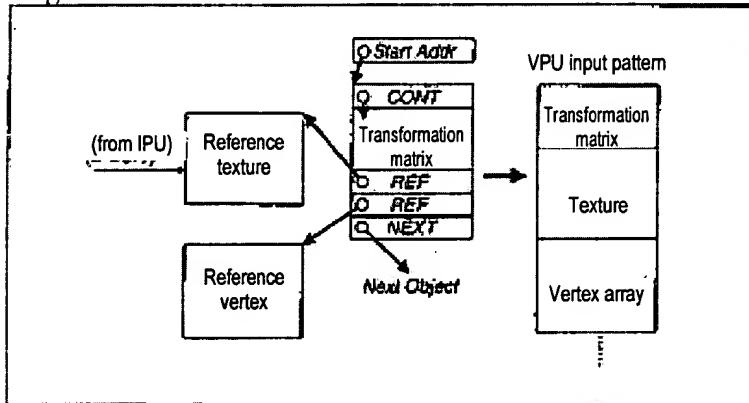
These computation resources are provided in order to eliminate the bottlenecks discussed in the preceding section. To this end, each unit has a number of built-in mechanisms.

#### 6.1 Capacity for computing coordinates

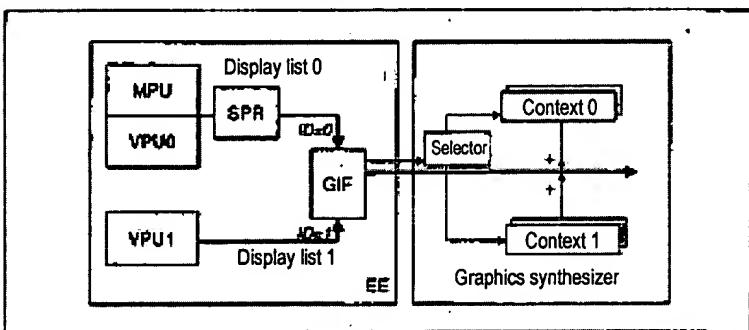
In order to solve the coordinate computation bottleneck, the two VPUs are of a form that allows

them to access the same rendering engine by time division. Thus an arbiter known as a GIF is provided at the output stage of the VPU (Figure 10). On the other hand, the rendering engine is able to hold multiple rendering contexts (graphics contexts) internally, and selects which context to use according to the context ID that is added to the display list (sequence of rendering instructions). If the context ID to be used by VPU0 and VPU1 is agreed to beforehand, display lists from VPU0 and VPU1 can be sent mixed together. Photo 2 shows an image generated by VPU1, and Photo 3 shows a

<Figure 9> Source chain DMA



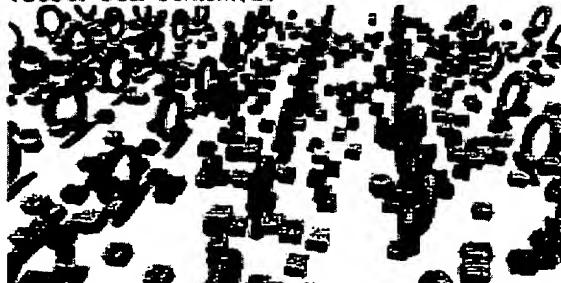
<Figure 10> Graphics context



<Photo 2> Dual Context (1)



<Photo 3> Dual Context (2)



superimposed image generated asynchronously by VPU0 while the bus to GS is empty.

The two VPUs are not positioned symmetrically. Of the two VPUs, VPU1 is devoted to fixed coordinate computation. By contrast, VPU0 is closely associated with the MPU and is devoted to physical calculations and other non-fixed processing. Taking the example of a hand-to-hand fighting game, objects (main characters, etc.) for which the calculation of complex action is required are assigned to VPU0, while objects that are simple but for which a number of polygons is

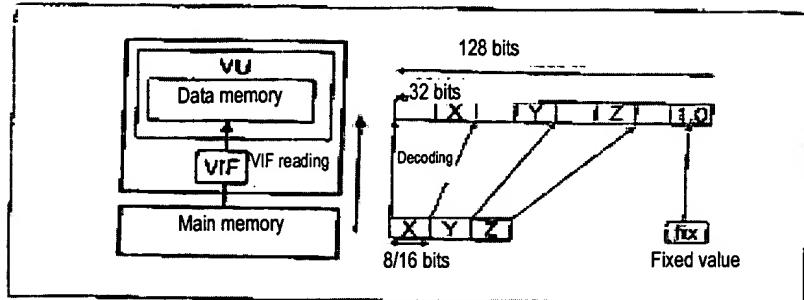
required (such as buildings in the background) are assigned to VPU1.

## 6.2 Memory access

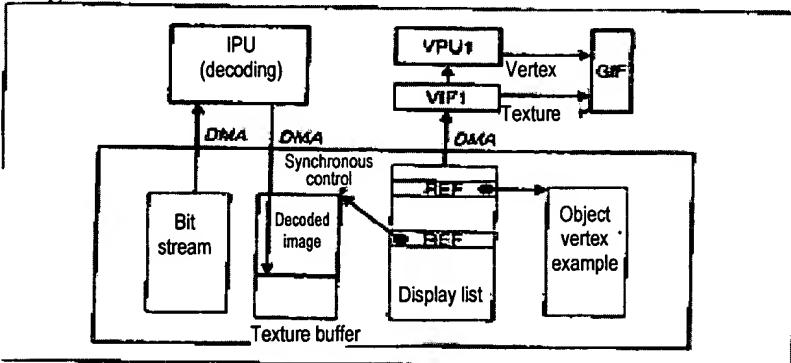
Bottlenecks in the memory system can be dealt with by holding the data in compressed form. In other words, if the data is expanded upon being read from the memory, not only can the memory capacity be economized, but the memory access speed can be improved as well.

Data that puts demands on the memory is mainly high-resolution texture data (image data for decorating polygon surfaces) or model data (three-dimensional coordinate data describing the shape of physical objects). Data of this type is kept in the main memory at all times in compressed form, and every time it is used it is first expanded and generated. Since today the operating clock of processors is sufficiently faster than the latency of the memory system, data can be sufficiently expanded in the cycles in the background of the time for reading from the memory. The IPU is used for expanding texture data. Data expanded by the IPU is assembled into part of the DMA chain and is sent to the rendering engine in the sequence in which it is expanded (Figure 11). What is used for expanding model data is the VPU interface unit (VIF) that is provided at the front end of the VPU (Figure 12).

<Figure 11> Unpacking coordinate values



<Figure 12> Online texture



### 6.3 Context switching

In order to avoid unnecessary context switching, dedicated units are pre-assigned to tasks (processes) that will always be needed. For example, the expansion of images is mandatory for the playback of moving pictures and the playback of fine textures, and thus it is better to have a dedicated unit for these tasks. Moreover, since the unit can be specialized for image expansion, it can be simpler, smaller, and faster-mounted than an ordinary processor. Also, this relieves the main processor of one extra task, correspondingly reducing the number of times context switching and interrupts take place.

This is one reason why it was decided to have two VPUs in parallel. If the goal were to simply have greater computation capacity, then instead of using two VPUs with four parallel SIMDs, the same result could be achieved with a

VLIW having eight parallel SIMDs, or with a single multiple instruction multiple data (MIMD) processor with four parallel data and parallel instructions. Indeed, in terms of memory efficiency, the latter choice might be better. But when game programs are analyzed, it becomes clear that completely different time schedules govern the part that calculates the behavior of the characters who appear from external input ("the characters who appear" is not limited to people but also applies to the movement of enemy weapons in shooting games or vehicles in racing games), and the part that responds to this and actually calculates the coordinate transformations and lighting processing. For the former, it suffices if calculations are completed in video frame (1/60 second) units, but the latter is more detailed and requires that calculations be done synchronized to the rendering pipeline, in

polygon rendering units. In a real-time system it will not do to use a single processor for such jobs that may look similar but actually have different time axes. The feature adopted to solve this problem is that, of the two VPUs, VPU0 does macro calculations synchronized to the video rate, while VPU1 can operate synchronized to the rendering engine. Thus VPU1 has a direct path to the rendering engine, bypassing the external bus. Conversely, VPU0 has a processor connection to the MPU so as to make it easy to carry out complicated processing programming.

Of course, it is only because the system is optimized for entertainment and multi-media applications that we can pull off this trick of being able to determine the necessary tasks and processes and take the right action. If, for sake of maintaining generality, the expansion of moving pictures were done by a SIMD expansion instruction of the main processor, there would be contention with the processor for calculating coordinates using other SIMD instructions, and registers would have to be saved and the floating-point calculation pipeline would be disrupted, causing great confusion.

### (7) Performance

The system's overall processing performance is presented in Table 3. The speed of coordinate transformations and perspective transformations of vertex coordinates is an important index of computer graphics processing, and here vertices can

&lt;Table 3&gt; Three-dimensional graphics processing performance

Processing item	Throughput
Effective bus bandwidth	2.0 Gbyte/sec
Floating point multiply-and-add calculation	6 GFLOPS
3D CG coordinate transform + perspective transform	
Without lighting processing	66 M polygon/sec
Parallel light source	38 M polygon/sec
Parallel light source and fog effect processing	36 M polygon/sec
Bezier curved-surface patch	15 M polygon/sec
Image decoding (macro block decoding)	150 M pixel/sec

be generated at up to 66 M polygon/second.

#### (8) Conclusion

As discussed above, the Emotion Engine has features that set it apart from ordinary so-called general-purpose processors.

First, while most processors pursue instruction-level parallelism (ILP), in the Emotion Engine, parallelism on the macro level is adopted to a greater extent. Its individual units have no ILP beyond the SIMD level. This is because, in terms of performance including the memory system as well, complex ILP does not promise effects commensurate with the scale of circuitry to be adopted.

By doing so, silicon is freed up to be allocated to vector units. What is popular in microprocessors today is to allocate much of the area of the chip to two-dimensional caches, and it was actually a high-risk adventure to boldly put VPUs on the chip instead of caches. To not want to do what others are doing is in Sony's genes, and as an actual problem, by simply doing the same, we could not outshine the many microprocessors that would precede ours. Above all, seen from the perspective of a well thought-out program, higher peak performance could be realized by having computing elements arranged in parallel than by having a copious cache. Game program

wizards prefer systems that have possibilities, even if they also have difficulties.

Instead, we are devoting many resources to data flow control to make full use of the computing elements. In order to use multiple computing elements efficiently, we must not create "wait" states in the units that control their operation. Thus the design focuses on allowing the units to perform memory access and computation in parallel, with data sent and received from the main memory automatically via the DMAC. This solves the problem of lockup that occurs when, in passing data between units, a large main memory is allocated and the intermediate buffers fill up.

With computers becoming cheaper and more popular, the day in which they were enshrined with awe as magic boxes that can do anything is becoming a thing of the past. Already, we are no longer living in an age in which general-purpose computers are used in great numbers. Given this, most computers will soon be systems that are intensely aware of individual uses and specific applications. In design too, we will be following the application model, in which it is the application that determines the architecture of the processor. The existence of a specific application image is very important in scenes where functions are trimmed away. Generally, it is more difficult to delete functions than to add on new

ones, and cutting them away requires courage. The Emotion Engine may be seen as an extreme embodiment of this point, in which resources are concentrated on calculation resources even at the expense of deleting other functions. Although at the present time a certain kind of performance is being hammered out, we do not yet have a final conclusion on whether this is the right approach to take. For that, we will have to wait until actual game titles hit the market. And those who will decide the answer are not computer scientists, or middle-age men who read economic newspapers, or securities analysts. The final arbiter is the market: children who save up their allowance and eagerly wait for the next-generation game machine to come on the market.

#### References

- [1] Itaru Motooka, Takashi Sugano, et al.: "VLSI Computers", published by Iwanami Shoten, 1985.
- [2] David A. Patterson, John L. Hennessy: *Computer Organization & Hardware/ Software Interface*, Morgan Kaufmann Publishers, 1994.
- [3] J. Foley, et al.: *Computer Graphics*, Addison-Wesley Publishing, 1990.
- [4] K. Kutaragi, et al.: A microprocessor with a 128b CPU, 10 Floating-Point MACs, 4 Floating-Point Dividers, and an MPEG2 Decoder, ISSCC digest, p. 256, 1999.
- [5] F. M. Raam, et al.: A High Bandwidth superscalar Microprocessor for Multimedia Applications, ISSCC digest, p. 258, 1999.

- Masakazu Suzuoki  
Sony Computer Entertainment  
suzu@rd.scei.sony.co.jp

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**